Vector Semantics

Giovanni Colavizza

Text Mining
Amsterdam University College

February 22, 2021

Announcements

- **Reading assignment 1 deadline: 21/02, 23:59**
- **Individual assignment 1 deadline: 24/02, 23:59**

Overview

1. Vector semantics
2. Matrix representations
3. Calculating vectors
4. Similarity measures
5. Evaluation

Vector semantics

The quest for meaning

- The realm of *lexical semantics*.
- A **lemma** can be associated with multiple **word senses** (e.g., "mouse (N)".
- The many facets of 'meaning':
  - *Propositional synonymity*: two words (senses) are synonyms if the truth conditions of a sentence does not change when we swap them.
  - *Word similarity*: some features are shared, but no synonyms. E.g., 'cat' and 'dog'.
  - *Word relatedness*: no features are shared, but there is a relationship. E.g., 'water' and 'bottle'.
  - Semantic frames or *topics*: topical structure in documents. E.g., 'sport' and 'politics'.
  - *Connotations*, e.g., sentiment (positive, negative), tone (formal or not).

Denotational vs distributional approaches

- Denotational approach: define (dictionary) meaning then apply definition. Meaning as dictionary index.
- Distributional approach: look at data to come up with meaning. **Distributional hypothesis**: "the amount of meaning difference between two words corresponds roughly to the amount of difference in their environments" (contexts of appearance). Harris, 1954.

Vector Semantics

“The meaning of a word is its use in the language.” Wittgenstein, 1953.

1. **???** *is best when cooked just right.*
2. *I prefer my* **???** *with abundant tomato sauce.*
3. *I eat* **???** *for lunch.*
4. *Would you like tomato sauce on your pizza?*
5. *We went to a pizza place for lunch.*
6. *Pizza should not be too cooked: it burns!*

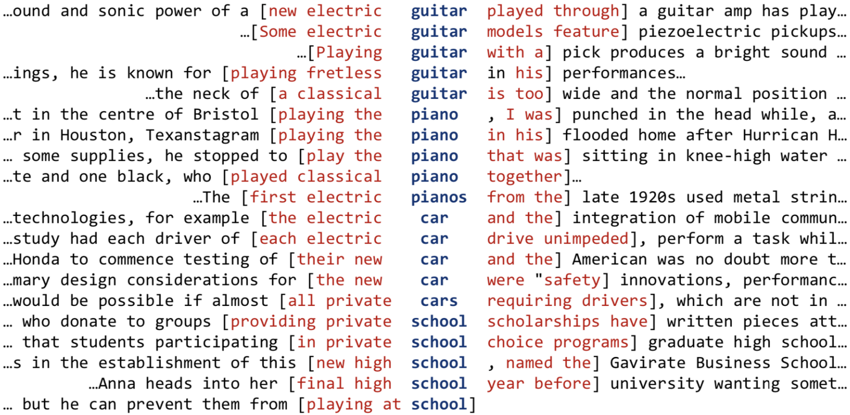
Can you guess **???**

Vector Semantics

"The meaning of a word is its use in the language." Wittgenstein, 1953.

1. *Pasta is best when cooked just right.*
2. *Pizza should not be too cooked: it burns!*
3. Vector semantics combines two intuitions:
   - **Distributional approach**: define a word by the contexts it occurs into.
   - **Vectorize it**: use vectors to represent word meaning, as a point in space.
4. **Feature engineering** for NLP: word vectors are increasingly used as features for other tasks.
5. (Word) vectors are usually referred as (word) **embeddings** in modern neural network literature.

Example

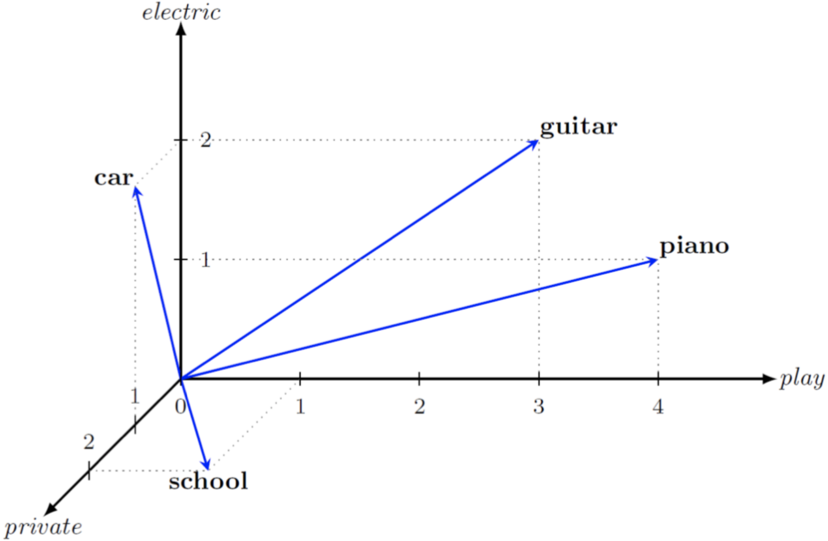


```
…ound and sonic power of a [new electric    guitar  played through] a guitar amp has play…
                         …[Some electric    guitar  models feature] piezoelectric pickups…
                              …[Playing     guitar  with a] pick produces a bright sound …
…ings, he is known for [playing fretless    guitar  in his] performances…
                ...the neck of [a classical guitar  is too] wide and the normal position …
…t in the centre of Bristol [playing the    piano   , I was] punched in the head while, a…
…r in Houston, Texanstagram [playing the    piano   in his] flooded home after Hurrican H…
… some supplies, he stopped to [play the    piano   that was] sitting in knee-high water …
…te and one black, who [played classical    piano   together]…
                     …The [first electric   pianos  from the] late 1920s used metal strin…
…technologies, for example [the electric     car    and the] integration of mobile commun…
…study had each driver of [each electric     car    drive unimpeded], perform a task whil…
…Honda to commence testing of [their new     car    and the] American was no doubt more t…
…mary design considerations for [the new     car    were "safety] innovations, performanc…
…would be possible if almost [all private    cars   requiring drivers], which are not in …
… who donate to groups [providing private   school  scholarships have] written pieces att…
… that students participating [in private   school  choice programs] graduate high school…
…s in the establishment of this [new high   school  , named the] Gavirate Business School…
           …Anna heads into her [final high school  year before] university wanting somet…
… but he can prevent them from [playing at school]
```

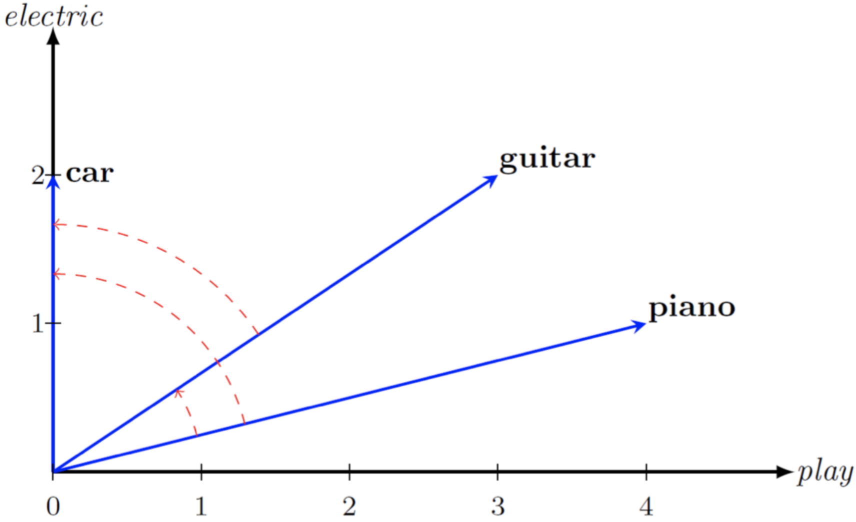
Example

	play	electric	classical	private	high	...	the	new
guitar	3	2	1	0	0	...	0	1
piano	4	1	1	0	0	...	4	0
car	0	2	0	1	0	...	4	2
school	1	0	0	2	2	...	1	1

Example



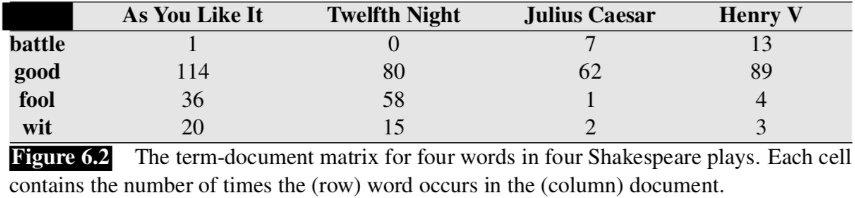
Example



Matrix representations

Word-Document matrix

- We have a set of documents D and a vocabulary V. X is a $|V| \times |D|$ matrix with word occurrences in documents.



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Figure 6.2 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.

Credit: J&M, ch. 6.

Word-Document matrix

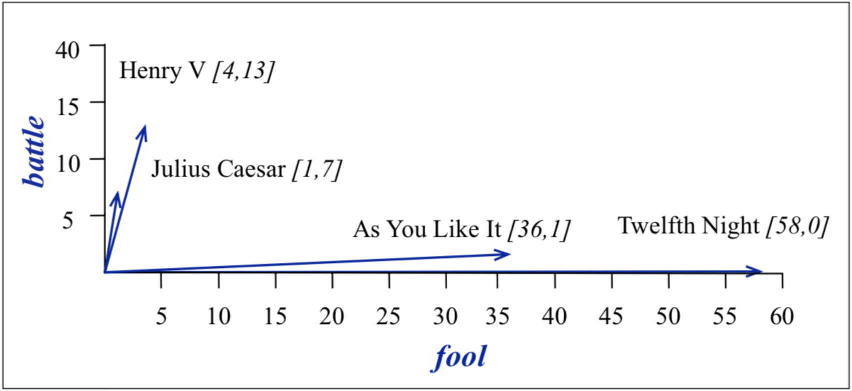
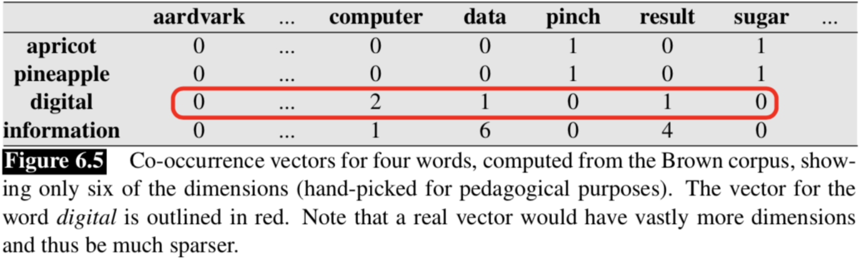


Figure 6.4 A spatial visualization of the document vectors for the four Shakespeare play documents, showing just two of the dimensions, corresponding to the words *battle* and *fool*. The comedies have high values for the *fool* dimension and low values for the *battle* dimension.

Credit: J&M, ch. 6.

Word-Context matrix

- We have a set of words V and a set of contexts they occur into C, taken from our corpus of documents. X in this case is a $|V| \times |C|$ matrix with word occurrences in contexts.
- The most intuitive context are co-occurrences with other words in V, within a certain **window**. In this case, X would be a $|V| \times |V|$ matrix.



	aardvark	...	**computer**	**data**	**pinch**	**result**	**sugar**	...
apricot	0	...	0	0	1	0	1	
pineapple	0	...	0	0	1	0	1	
digital	0	...	2	1	0	1	0	
information	0	...	1	6	0	4	0	

Figure 6.5 Co-occurrence vectors for four words, computed from the Brown corpus, showing only six of the dimensions (hand-picked for pedagogical purposes). The vector for the word *digital* is outlined in red. Note that a real vector would have vastly more dimensions and thus be much sparser.

Credit: J&M, ch. 6.

Word-Context matrix

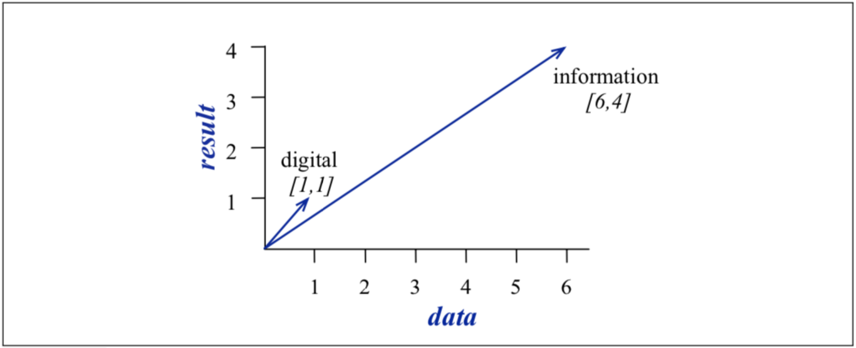


Figure 6.6 A spatial visualization of word vectors for *digital* and *information*, showing just two of the dimensions, corresponding to the words *data* and *result*.

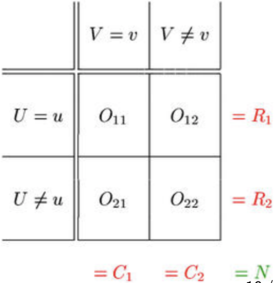
Credit: J&M, ch. 6.

Types of co-occurrences

- **Surface co-occurrence**:
  - a contextual word co-occurs with the target word as many times as the former appears in a collocational span (window) surrounding the latter.
  - The span may be symmetric ([-5w, +5w]) or asymmetric ([-5w, 0]).
- **Textual co-occurrence**:
  - words co-occur if they appear in the same text segment (e.g., a sentence, a paragraph, a web page ...).
  - It usually does not matter how many times each word occur in each document.
- **Syntactic co-occurrence**:
  - count word co-occurrences in a specific syntactic relation (e.g., verb-object, adjective-noun ...).

Contingency tables

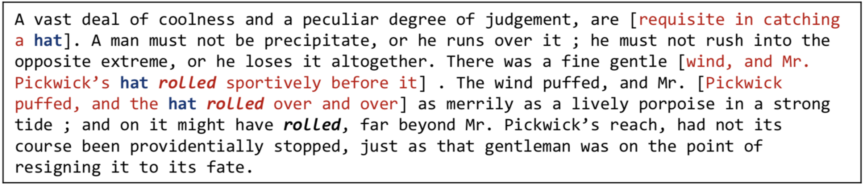
- Tabular representation of the observed frequencies between the variable whose values are reported in rows and the variable whose values are reported in columns.
- Intermediate step for some calculations we will see.
- If u is our target word and v is a contextual word:
  - O_{11} – observed frequency of u and v (i.e., $f(u, v)$).
  - R_1, R_2, C_1, C_2 – marginal frequencies.
  - R_1 – absolute frequency of u (i.e. $f(u)$).
  - C_1 – absolute frequency of v (i.e. $f(v)$).
  - $N = O_{11} + O_{12} + O_{21} + O_{22}$ – sample size.



	$V = v$	$V \neq v$	
$U = u$	O_{11}	O_{12}	$= R_1$
$U \neq u$	O_{21}	O_{22}	$= R_2$
	$= C_1$	$= C_2$	$= N$

Contingency tables – Surface co-occurrences

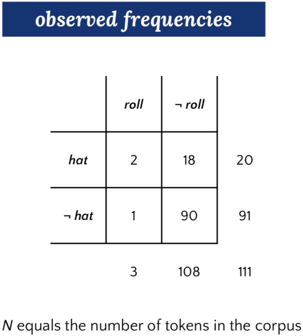
- w1 = “hat”; w2 = “roll”.
- Collocational span: ±4 words (i.e. [-4w, +4w]).
- Spans cannot cross sentence boundaries.



A vast deal of coolness and a peculiar degree of judgement, are [requisite in catching a **hat**]. A man must not be precipitate, or he runs over it ; he must not rush into the opposite extreme, or he loses it altogether. There was a fine gentle [wind, and Mr. Pickwick’s **hat** ***rolled*** sportively before it] . The wind puffed, and Mr. [Pickwick puffed, and the **hat** ***rolled*** over and over] as merrily as a lively porpoise in a strong tide ; and on it might have ***rolled***, far beyond Mr. Pickwick’s reach, had not its course been providentially stopped, just as that gentleman was on the point of resigning it to its fate.

Contingency tables – Surface co-occurrences

- w1 = “hat”; w2 = “roll”.
- Collocational span: ±4 words (i.e. [-4w, +4w]).
- Spans cannot cross sentence boundaries.



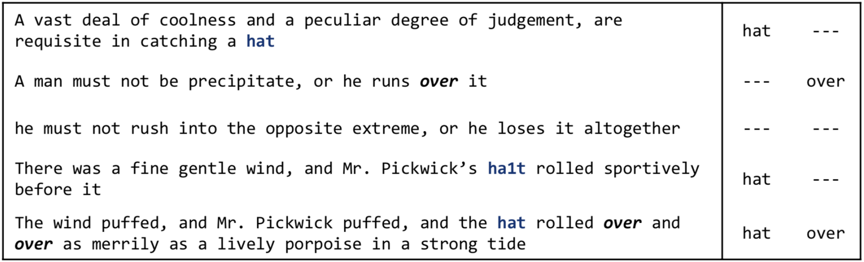
observed frequencies

	roll	*¬ roll*	
hat	2	18	20
¬ hat	1	90	91
	3	108	111

NOTE: *N* equals the number of tokens in the corpus

Contingency tables – Textual co-occurrences

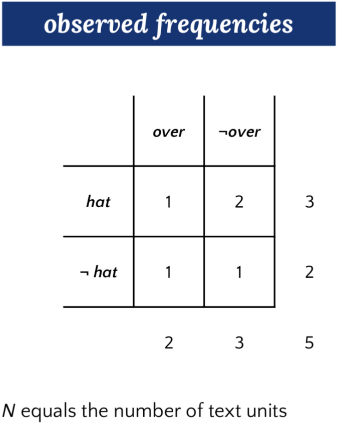
- w1 = “hat”; w2 = “over”.
- Unit is sentence (multiple occurrence is the same unit are ignored).



A vast deal of coolness and a peculiar degree of judgement, are requisite in catching a **hat**	hat	---
A man must not be precipitate, or he runs ***over*** it	---	over
he must not rush into the opposite extreme, or he loses it altogether	---	---
There was a fine gentle wind, and Mr. Pickwick’s **ha1t** rolled sportively before it	hat	---
The wind puffed, and Mr. Pickwick puffed, and the **hat** rolled ***over*** and ***over*** as merrily as a lively porpoise in a strong tide	hat	over

Contingency tables – Textual co-occurrences

- w1 = “hat”; w2 = “over”.
- Unit is sentence (multiple occurrence is the same unit are ignored).



observed frequencies

	over	*¬over*	
hat	1	2	3
¬ hat	1	1	2
	2	3	5

NOTE: *N* equals the number of text units

Calculating vectors

Families of vectors

- **Sparse vectors**: many zero values and high-dimensional spaces. E.g., weighted co-occurrence matrices (this class).
- **Dense vectors**: no zero values and smaller-dimensional spaces.
  - Dimensionality reduction (Latent Semantic Analysis or truncated Singular Value Decomposition, Principal Component Analysis, Non-negative Matrix Factorization, and many more): mostly we skip, there is a little in the next lab.
  - Neural-networks (Skigp-gram, CBOW, GloVe): next class.

(Better) quantifying association

Raw co-occurrence frequency is often not the optimal measure of association between a word and a context:

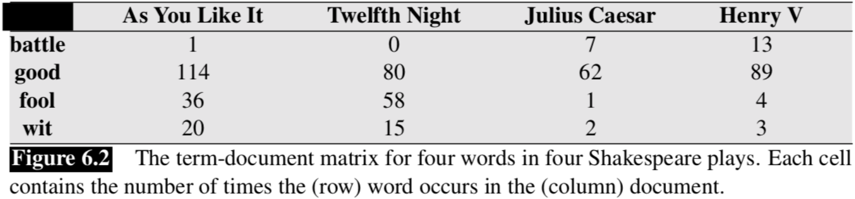
- we need a way to estimate to what extent a context word is particularly informative about a target word;
- frequencies are very skewed.
- A couple of solutions: **tf-idf** and **PPMI**.

Term frequency - Inverse document frequency

- Tf-idf is the standard weighting scheme for term-document matrices.
- It is likely the most used weighting scheme in Information Retrieval.
- **Term frequency** $tf(t,d)$ = the number of times term t occurs in document d. M any variants exist (e.g., using a log transform). It accounts for how frequent t is within the document collection.
- **Inverse document frequency** $idf(t) = log\left(\frac{|D|}{|D_t|}\right)$; where D is the collection of documents and D_t is the subset where term t appears once or more. It accounts for how 'discriminative' t is with respect to the document collection.

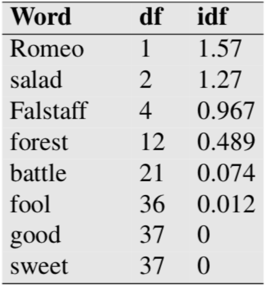
$$tfidf(t,d) = tf(t,d) \times idf(t)$$

Term frequency - Inverse document frequency



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

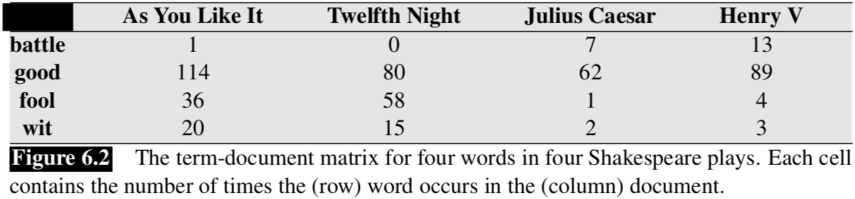
Figure 6.2 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.



Word	df	idf
Romeo	1	1.57
salad	2	1.27
Falstaff	4	0.967
forest	12	0.489
battle	21	0.074
fool	36	0.012
good	37	0
sweet	37	0

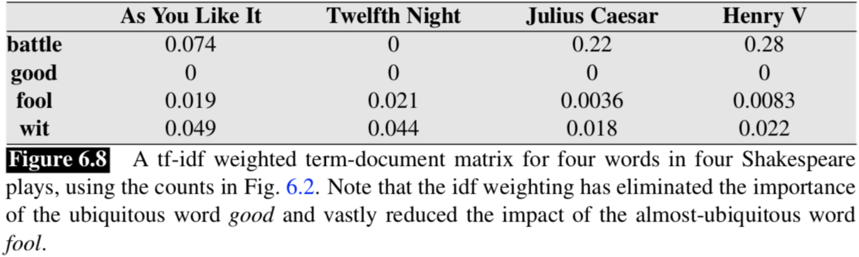
Credit: J&M, ch. 6.

Term frequency - Inverse document frequency



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	0	7	13
good	114	80	62	89
fool	36	58	1	4
wit	20	15	2	3

Figure 6.2 The term-document matrix for four words in four Shakespeare plays. Each cell contains the number of times the (row) word occurs in the (column) document.



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	0.074	0	0.22	0.28
good	0	0	0	0
fool	0.019	0.021	0.0036	0.0083
wit	0.049	0.044	0.018	0.022

Figure 6.8 A tf-idf weighted term-document matrix for four words in four Shakespeare plays, using the counts in Fig. 6.2. Note that the idf weighting has eliminated the importance of the ubiquitous word *good* and vastly reduced the impact of the almost-ubiquitous word *fool*.

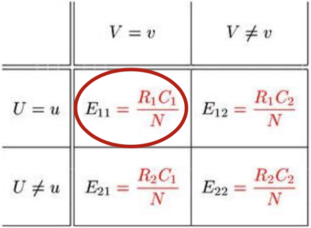
Credit: J&M, ch. 6.

(Positive) Pointwise Mutual Information

Intuition: in order to discriminate between informative and uninformative word-context associations, let us **take the expected frequency into account as a baseline**.

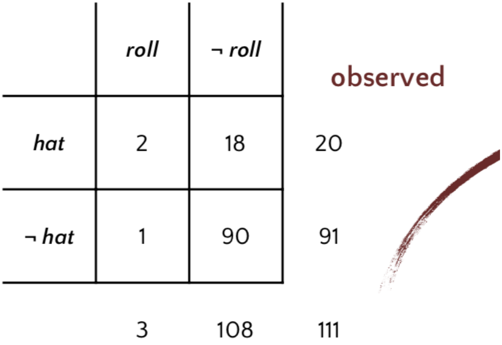
- The **expected frequency** of a (word, context) pair is a measure of how often a word would occur in a context if the two linguistic entities were statistically independent (i.e., if they were occurring by chance).
- The expected frequency can be estimated from the marginals in the contingency table:

$$E_{11} = \frac{f(u)f(v)}{N}$$



	$V = v$	$V \neq v$
$U = u$	$E_{11} = \frac{R_1 C_1}{N}$	$E_{12} = \frac{R_1 C_2}{N}$
$U \neq u$	$E_{21} = \frac{R_2 C_1}{N}$	$E_{22} = \frac{R_2 C_2}{N}$

(Positive) Pointwise Mutual Information



	roll	*¬ roll*	
hat	2	18	20
¬ hat	1	90	91
	3	108	111

observed

(Positive) Pointwise Mutual Information

	roll	*¬ roll*	
hat	2	18	20
¬ hat	1	90	91
	3	108	111

observed

expected

	roll	*¬ roll*
hat	0.54	19.46
¬ hat	2.46	88.54

(Positive) Pointwise Mutual Information

- Mutual Information provides a measure of independence of two random variables X and Y:

$$MI(X, Y) = \sum_{x \in X} \sum_{y \in Y} P(x, y) log \frac{P(x, y)}{P(x)P(y)}$$

- Pointwise Mutual Information is the part related to two outcomes x and y:

$$PMI(x, y) = log \frac{P(x, y)}{P(x)P(y)}$$

- Us, we are interested in a word-context pair, w and c:

$$PMI(w, c) = log \frac{P(w, c)}{P(w)P(c)}$$

Positive Pointwise Mutual Information

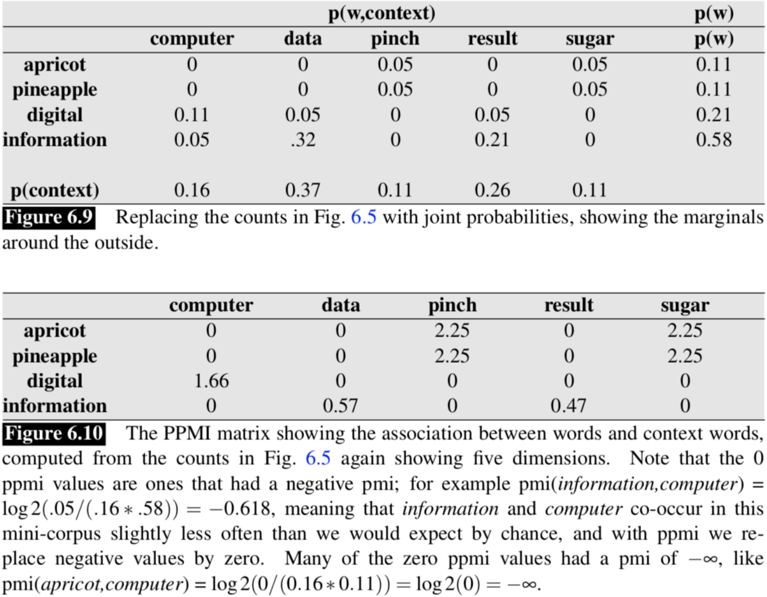
- We are not interested in joint events more unlikely than independent ones, thus we usually just consider the positive values of **PPMI**:

$$PPMI(w, c) = max\left(0, log \frac{P(w, c)}{P(w)P(c)}\right)$$

- Many variants to account for minor issues:
  - **Positive Local Mutual Information**: deals with the tendency of PPMI to emphasize rare events over frequent ones:

$$PLMI(w, c) = max\Big(0, f(w, c) \times PMI(w, c)\Big)$$

PPMI



	p(w,context)					p(w)
	computer	**data**	**pinch**	**result**	**sugar**	**p(w)**
apricot	0	0	0.05	0	0.05	0.11
pineapple	0	0	0.05	0	0.05	0.11
digital	0.11	0.05	0	0.05	0	0.21
information	0.05	.32	0	0.21	0	0.58
p(context)	0.16	0.37	0.11	0.26	0.11	

Figure 6.9 Replacing the counts in Fig. 6.5 with joint probabilities, showing the marginals around the outside.

	computer	**data**	**pinch**	**result**	**sugar**
apricot	0	0	2.25	0	2.25
pineapple	0	0	2.25	0	2.25
digital	1.66	0	0	0	0
information	0	0.57	0	0.47	0

Figure 6.10 The PPMI matrix showing the association between words and context words, computed from the counts in Fig. 6.5 again showing five dimensions. Note that the 0 ppmi values are ones that had a negative pmi; for example pmi(*information,computer*) = $\log 2(.05/(.16 * .58)) = -0.618$, meaning that *information* and *computer* co-occur in this mini-corpus slightly less often than we would expect by chance, and with ppmi we replace negative values by zero. Many of the zero ppmi values had a pmi of $-\infty$, like pmi(*apricot,computer*) $= \log 2(0/(0.16 * 0.11)) = \log 2(0) = -\infty$.

Credit: J&M, ch. 6.

Similarity measures

Comparing vectors: the dot product

- Now we know how to calculate the first-order associations between words and how to use this information to create a *distributional representation* of each word.
- **Similarity measures** can be used to quantify the distance between two vectors in a space, and this can be used to estimate how similar the represented words are.
- Most vector similarity measure are based on the **dot (inner) product**:

$$\vec{u} \cdot \vec{v} = \sum_{i=1}^{d} u_i v_i = u_1 v_1 + u_2 v_2 + \cdots + u_d v_d$$

Comparing vectors: Euclidean distance

- When used as a similarity metric, the dot product has a problem: it favors vectors with higher values (e.g., frequencies).
- It is the same issue you have with the Euclidean norm:

$$||\vec{v}|| = \sqrt{\sum_{i=1}^{d} v_i^2}$$

- from which the Euclidean distance stems:

$$d(\vec{u}, \vec{v}) = ||\vec{u} - \vec{v}|| = \sqrt{\sum_{i=1}^{d} (u_i - v_i)^2}$$

- A similarity measure that is sensitive to frequency can sometimes work, yet other times it is the direction of vectors which is more important.

Comparing vectors: Cosine distance

- A solution is to use the **cosine of the angle between the two vectors**:

$$cosine(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{||\vec{u}||||\vec{v}||} = \frac{\sum_{i=1}^{d} u_i v_i}{\sqrt{\sum_{i=1}^{d} u_i^2}\sqrt{\sum_{i=1}^{d} v_i^2}}$$

- The cosine ranges between 1 and -1, taking value 0 for orthogonal vectors. Due to the fact that PPMIs and other frequencies are always non-negative, cosine ranges from 0 to 1 (identically directed vectors).

Cosine

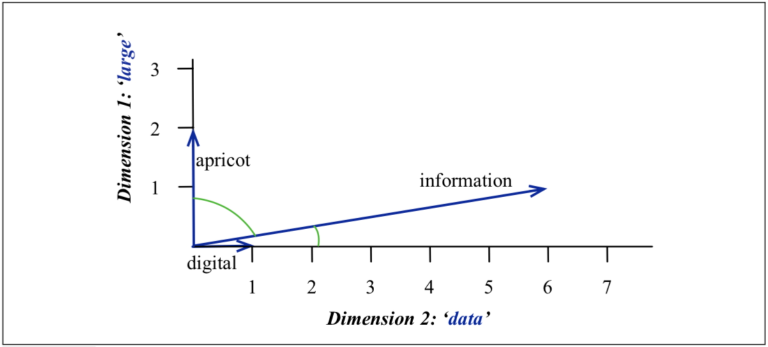


Figure 6.7 A graphical demonstration of cosine similarity, showing vectors for three words (*apricot*, *digital*, and *information*) in the two dimensional space defined by counts of the words *data* and *large* in the neighborhood. Note that the angle between *digital* and *information* is smaller than the angle between *apricot* and *information*. When two vectors are more similar, the cosine is larger but the angle is smaller; the cosine has its maximum (1) when the angle between two vectors is smallest (0°); the cosine of all other angles is less than 1.

Credit: J&M, ch. 6.

Comparing vectors: Probabilistic measures

- The Euclidean and cosine distances are geometric measures. Sometimes is more convenient to see vectors as probability distributions (after appropriate normalization).
- Many measures exists to compare two probability distributions, for example the Kullback-Leibler divergence (non-simmetric) or the (simmetric) Jensen-Shannon divergence:

$$D_{KL}(\vec{u}||\vec{v}) = \sum_{i=1}^{d} u_i log\left(\frac{u_i}{v_i}\right)$$

$$D_{JS}(\vec{u}||\vec{v}) = \frac{1}{2}D_{KL}(\vec{u}||\vec{m}) + \frac{1}{2}D_{KL}(\vec{v}||\vec{m})$$

where $m = \frac{(\vec{u}+\vec{v})}{2}$

Evaluation

Association/Similarity

Two words or a word and a context, may have two kinds of associations:

- **Syntagmatic associations** (first-order co-occurrence): how much two words appear one next to the other.
  1. E.g., the association between a verb and its typical complements;
  2. what is represented in a co-occurrence matrix.
- **Paradigmatic associations** (second-order co-occurrence) is similarity of context: how similar are the neighbors of the two words.
  1. E.g., the association between two synonyms, or “wrote”, “said”, “remarked”.
  2. what is estimated by calculating (first-order) vectors similarity.

Evaluation of vectors

- The most common evaluation is to test their performance on similarity tasks.
- Correlation between algorithm and human word similarity ratings:
    1. *WordSim-353*: 353 noun pairs rated on a 0-10 scale. sim("plane", "car") = 5.77.
    2. *SimLex-999*: similarity of noun, adjectives and adjectives pairs. *Assignment 3*.
- Taking *TOEFL* multiple-choice vocabulary tests
    1. Levied is closest in meaning to: imposed, believed, requested, correlated.
- Judgments in context
    1. *Stanford Contextual Word Similarity* (SCWS) dataset: human judgments on 2,003 pairs of nouns, verbs, and adjectives in their sentence context.